PCT/EP00/11710

Europäisches Patentamt | European Patent Office | Office européen des brevets

NL 000643

EP00/11710

4

(44)

Bescheinigung | Certificate | Attestation

09/913670

2/6

Die angehefteten Unterlagen stimmen mit der ursprünglich eingereichten Fassung der auf dem nächsten Blatt bezeichneten europäischen Patentanmeldung überein.

The attached documents are exact copies of the European patent application described on the following page, as originally filed.

Les documents fixés à cette attestation sont conformes à la version initialement déposée de la demande de brevet européen spécifiée à la page suivante.

Patentanmeldung Nr. | Patent application No. | Demande de brevet n°

99204369.5

Der Präsident des Europäischen Patentamts;
Im Auftrag

For the President of the European Patent Office

Le Président de l'Office européen des brevets
p.o.

I.L.C. HATTEN-HECKMAN

DEN HAAG, DEN
THE HAGUE, 11/12/00
LA HAYE, LE

Europäisches Patentamt
European Patent Office
Office européen des brevets

Blatt 2 der Bescheinigung
Sheet 2 of the certificate
Page 2 de l'attestation

Anmeldung Nr.:
Application no.: 99204369.5
Demande n°:

Anmeldetag:
Date of filing: 17/12/99
Date de dépôt:

Anmelder:
Applicant(s):
Demandeur(s):
Koninklijke Philips Electronics N.V.
5621 BA Eindhoven
NETHERLANDS

Bezeichnung der Erfindung:
Title of the invention:
Titre de l'invention:
Second generation DSP software for picture rate conversion

In Anspruch genommene Priorität(en) / Priority(ies) claimed / Priorité(s) revendiquée(s)

Staat: State: Pays:	Tag: Date: Date:	Aktenzeichen: File no. Numéro de dépôt:

Internationale Patentklassifikation:
International Patent classification:
Classification internationale des brevets:
/

Am Anmeldetag benannte Vertragstaaten:
Contracting states designated at date of filing: AT/BE/CH/CY/DE/DK/ES/FI/FR/GB/GR/IE/IT/LI/LU/MC/NL/PT/SE/TR
Etats contractants désignés lors du depôt:

Bemerkungen:
Remarks:
Remarques:

Second generation DSP software for picture rate conversion

R.B. Wittebrood G. de Haan

Introduction

Advances in multimedia have led to a large range of video formats and, more specific, an increasing number of picture update frequencies. Therefor, mismatches between the, PC or TV, display rate and the video rate increasingly occur and conversions are necessary for interfacing. The complexity of the applied algorithms range from very simple, e.g. picture repetition, to highly advanced applying motion estimation and compensation techniques. The more sophisticated methods are necessary to prevent motion judder and motion blur. Dedicated consumer ICs for high quality picture rate conversion have been available for some years. Recently, algorithmic progress resulted in a first generation high quality picture rate conversion in software that runs real time on a DSP [1, 2]. Since then a number of significant algorithmic advances caused a considerably improved performance. The most relevant progress was made in the algorithm for motion estimation (ME). This paper presents the new ME algorithm, while we refer to [1] for the motion compensation and de-interlacing.

The new motion estimator

The ME is a further development of the concept described in [1], which aims at estimating motion on an object basis and uses sub-sampled images (4 × hor., 2 × ver.). Compared to [1], the parameters of the object motion models, $\vec{P}_o(n) = (s_x, s_y, d_x, d_y)$ are extended with *two additional coefficients*, d_x and d_y. As in [1], the parameter vector of an object is determined on a small set of "interesting" image parts, SI, focusing objects by giving different weights to different positions. High weights (> 1) are given to positions where the model resulted to be best in the previous segmentation, and low weights (< 1) are given to areas where competing models turned out to be better. The actual object parameter vector is a result of a log-search through the parameter vector space, optimising the error criterion that remained unchanged:

$$\epsilon(\vec{C}_o, n) = \sum_{\vec{x} \in SI} W_o(\vec{x}) \cdot |F_s(\vec{x}, n) - F_s(\vec{x} - \vec{C}_o(\vec{x}, n), n-1)| \quad (1)$$

where $F_s(\vec{x}, n)$ is the luminance value at position $\vec{x}$ of the *sub-sampled* image with index n, $C_o(\vec{x}, n)$ is the vector resulting from candidate model $\vec{C}_o(n)$ at position $\vec{x}$ and $W_o(\vec{x})$ is the weight at position $\vec{x}$

An important improvement occurred in the segmentation. Previously ([1]), the segmentation mask assigned the candidate object o with the lowest local *modified* cost function $\epsilon_o(\vec{X}, n)$ to the block $B(\vec{X})$:

$$\epsilon_o(\vec{X}, n) = \sum_{\vec{x} \in B(\vec{X})} |F_s(\vec{x}, n) - F_s(\vec{x} - \vec{D}_o(\vec{x}, n), n-1)| + P(\vec{X}, n) \quad (2)$$

A first modification is that we allowed a small deviation of the local motion vector from the object motion model, using a bias vector $\vec{B}(\vec{X}, n)$:

$$\vec{D}_o(\vec{X}, n) = \begin{pmatrix} s_x(o, n) + X d_x(o, n) \\ s_y(o, n) + Y d_y(o, n) \end{pmatrix} + \vec{B}(\vec{X}, n) \quad (3)$$

$\vec{D}_o(\vec{x}, n)$ is the motion vector determined by object o and the bias field $\vec{B}(\vec{X}, n)$. This allows more complex object motions to be modelled than with the 4 simple objects alone.

Another difference is that we added a penalty, $P(\vec{X}, n)$, to the error function in eq. (2), while the evaluated object vectors are reduced to a *subset* of the available object models, using values occurring in a spatio-temporal neighbourhood, adding their corresponding bias vectors and extending this set with a random candidate for the bias vector and a single object vector that *does not* occur in the neighbourhood. The penalties are selected smaller for the "neighbouring" candidates and higher for the randomly selected vectors, similar to what has been described in [3]. The effect of this modification is an increased spatio-temporal consistency of the vector field and a reduced operations count, as only a limited number of inconsistent vectors are evaluated.

The set of "interesting" image parts, or points of interest (POIs), is determined by selecting a fixed number of samples from the set of blocks for which the error criterion, eq. (1) is above a threshold T_{POI} for $\vec{C}_o(\vec{x}, n) = 0$. In figure 1 an example is given of the POIs and the segmentation mask. Figure 2 gives a block diagram of the new ME.

A refinement bias field improving an object motion estimator

G. de Haan and R.J. Wittebrood

Philips Research Laboratories, Television Systems Group,
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

A "bias" field for an earlier designed object based motion estimator is proposed, suitable for high-quality motion compensated scan-rate conversions. Compared to the prior art, the advantage of the resulting object motion estimator with bias field is that the object motion can be modelled with relatively simple motion models, while achieving high accuracy, as the bias fied describes the (small) deviations from the simple motion models used.

Keywords: **motion estimation, parametric motion models, image segmentation, scan rate conversion.**

1 Introduction

The motion estimator described in this paper was designed to be suitable for scan rate conversion, with a computational complexity suitable for consumer electronics application, i.e. comparable to [5, 6].

The most striking characteristic of the object motion estimator described earlier in [1], is that no effort is put in segmenting the image into objects prior to estimation of the model parameters. Basically, a relatively small number of interesting image parts is selected, and a number of parallel motion model parameter estimators is trying to optimize their parameters on this data set. As soon as one of the estimators is more successful than another in a certain number of interesting image parts, it is focused on those parts, whereas the remaining estimators focus on the other parts. In short: individual estimators try to conquer image parts from eachother, dividing the total image into "objects".

Fundamentally, such an *object-based* motion estimator[1] that wastes no effort in expensive segmentation of the image should be able to compete in operations count with a *block based* motion estimator, as one should expect less objects than blocks in realistic images. It is only in the assignment of image parts to objects that an effort is required comparable to the evaluation of candidate vectors on block basis. If the number of objects does not exceed the number of candidate vectors too much, the overhead of an object based motion estimator should be neglegible. It is assumed here that the motion per object can be described with fairly simple parametric models(, corrected with a local bias vector).

In the following subsections, we shall describe the motion model used, the estimation of the model parameters, the cost function used, and the new recursive segmentation of the image.

1.1 The motion model

To keep complexity low, the motion of each object o is described by a simple first order linear model that can only describe translation and scaling[2],

$$\vec{D}_o(\vec{x},n) = \begin{pmatrix} s_x(o,n) + xd_x(o,n) \\ s_y(o,n) + yd_y(o,n) \end{pmatrix} \qquad (1)$$

using $\vec{D}_o(\vec{x},n)$ for the displacement vector of object o at location $\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ in the image with index n.

1.2 Parameter estimation

Given a motion model, next its parameters need to be optimised for a given object in the image. As stationary image parts occur in almost every sequence, we

[1] An object, in the context of this estimator, refers to image parts that can be described with the same motion model. These image parts do not necessarily correspond to a single physical object in the scene.

[2] More complex parametric motion models have been proposed and can indeed be applied in combination with the proposed algorithm, but will be disregarded here, as we shall introduce a refinement that makes such complex models obsolete.

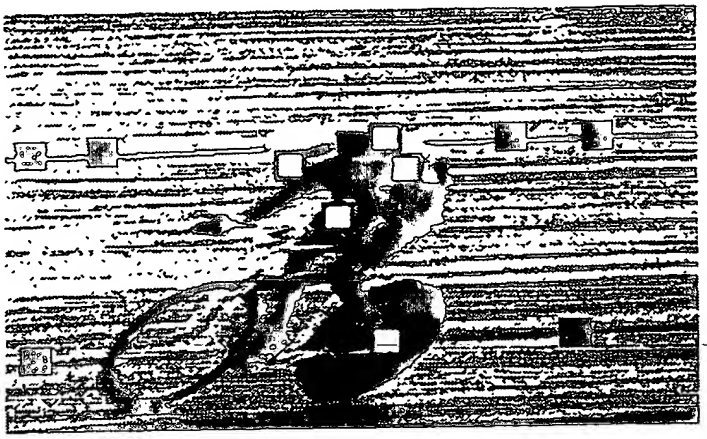

Figure 2: Example of selection of points of interest. The grey blocks are emphasized pixel blocks of interest in PE_1, the white ones are emphasized in PE_2.

- The pixel values are multiplied with a second weight factor smaller than 1, e.g. 0.1, in case the segmentation assigned the position to another parameter estimator and this estimator achieved low match errors.

Figure 2 gives an example of the selection of the pixel blocks of interest in an image with a single moving object and a moving background. The moving background of the image is object $o = 1$, and the bicyclist is object $o = 2$. Both parameter estimators are optimized on the same set containing the blocks of interest, but as soon as one estimator is selected in the segmentation to be best in an area, the pixel block of interest in that area is emphasized in the cost function. After a while, this converges to the situation illustrated, where one estimator focusses om the grey blocks and the other on the white pixel blocks in $SI(n)$.

More formally, the cost function is calculated according to:

$$\epsilon(\vec{C}_o, n) = \sum_{\vec{x} \in IS} W_o(\vec{x}) \cdot |F_s(\vec{x}, n) - F_s(\vec{x} - \vec{C}_o(\vec{x}, n), n-1)| \quad (5)$$

where $F_s(\vec{x}, n)$ is the luminance value at position $\vec{x}$ in a *sub-sampled* image with index n, and $C_o(\vec{x}, n)$ is the vector resulting from candidate model $\vec{C}_o(n)$ at position $\vec{x}$.

The sub-sampling effectively reduces the required memory bandwidth. Images are sub-sampled with a factor of four horizontally and a factor of two vertically on a field base, generating a sub-sampled image $F_s(n)$ from each original field $F(n)$. In order to achieve pixel accuracy on the original pixel grid of F, interpolation is required on the sub-sampling grid.

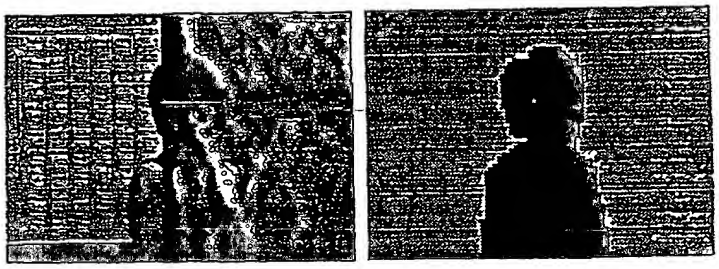

Figure 3: Picture and the resulting segmentation.

1.4 Recursive segmentation

The segmentation is the most critical step in the algorithm. Its task is to assign one motion model to each group of pixels. This is basically achieved by assigning the best matching model to each group of pixels (a block B, which is typically as large as 8×8 pixels on frame base).

For each block, a *block* match error, $\epsilon_o(\vec{X}, n)$ corresponding to each of the estimated parameter vectors, $\vec{P}_o$, can be calculated according to:

$$\epsilon_o(\vec{X}, n) = \sum_{\vec{x} \in B(\vec{X})} |F_s(\vec{x} + (1-\alpha)\vec{D}_o(\vec{x}, n), n) - F_s(\vec{x} - \alpha\vec{D}_o(\vec{x}, n), n-1)| \quad (6)$$

The temporal instance where this segmentation is valid is defined by α.

The straightforward equivalent of the full-search block matcher, would result if the segmentation mask $M(\vec{X}, n)$ assigned the object o with the lowest block match eror, ϵ_o, to the block $B(\vec{X})$. Such a segmentation is more dangerous when many objects occur in the image, since inconsistencies, similar to those of a full search block matcher, are not prevented.

Therefore, we adopted a recursive segmentation method that more closely resembles the strategy of the 3-D RS block matcher, i.e. use spatial and temporal predictions of the best $PE_o(n)$ and penalise chosing a $PE_o(n)$ that does not occur in the spatio-temporal neighbourhood. Formally, the segmentation mask $M(\vec{X}, n)$ assigns the object o with the lowest local *modified* cost function $\epsilon_o{}'(\vec{X}, n)$ to the block $B(\vec{X})$, where

$$\epsilon_o{}' = \epsilon_o + P(\vec{X}, n) \quad (7)$$

while $P(\vec{X}, n)$ is a penalty choosen according to the

6] G. de Haan, and P.W.A.C. Biezen. 'Sub-pixel motion estimation with 3-D recursive search block-matching', *Signal Processing: Image Communication 6*. 1994, pp.229-239. G. de Haan. J. Kettenis, and B. Deloore. 'IC for Motion Compensated 100Hz TV. with a Smooth Motion Movie-Mode', *IEEE Tr. on Consumer Electronics*. vol. 42. no. 2. May 1996. pp. 165-174.

7] G. de Haan and P.W.A.C. Biezen, 'Time-recursive de-interlacing for high-quality television receivers.' *Proc. of the Int. Workshop on HDTV and the Evolution of Television*. Taipei, Taiwan. November 1995. pp. 8B25–8B33.

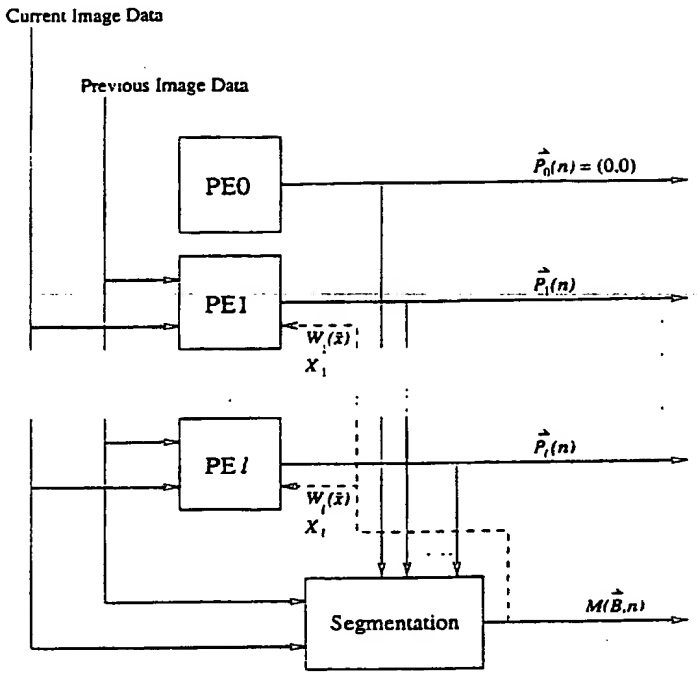

Figure 1: Block diagram of multiple parameter estimators and segmentation.

no estimation effort is required to make this available. The parameter vectors of additional objects, o, $o > 0$, are estimated separately, in parallel, by their respective parameter estimators (PE_o), as shown in Figure 1. Each PE_o updates a previously estimated parameter vector, after which the best parameter candidate vector, according to a cost function, is selected as the result parameter vector for that object.

Considering the four parameter model of eq. (1), the parameters of object o, $o > 0$. are regarded as a parameter vector $\vec{P}_o$:

$$\vec{P}_o(n) = \begin{pmatrix} s_x(o,n) \\ s_y(o,n) \\ d_x(o,n) \\ d_y(o,n) \end{pmatrix} \quad (2)$$

and we define our task as to select $\vec{P}_o(n)$ from a number of candidate parameter vectors $\vec{C}_o(n)$ as the one that has the minimal value of a cost function, to which we shall return lateron.

The candidates are generated much similar to the strategy exploited in [5, 6], i.e. take a prediction vector, add at least one update vector, and select the best candidate parameter vector according to an error criterion. Candidate parameter set $CS_o(n)$ contains three candidates $\vec{C}_o(n)$ according to:

$$CS_o(n) = \{\vec{C}_o(n) | \vec{C}_o(n) = \vec{P}_o(n-1) + m\vec{U}_o(n).$$
$$\vec{U}_o(n) \in US_o(n).$$
$$m = -1, 0.1\} \quad (3)$$

with update parameter $\vec{U}_o(n)$ selected from update parameter set $US_o(n)$:

$$US_o(n) = \left\{ \begin{pmatrix} i \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ i \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ i \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ i \end{pmatrix} \right\}.$$
$$(i = 1, 2, 4, 8, 16) \quad (4)$$

1.3 The cost function

Given the motion model and some candidate parameter sets, we need to select the best candidate, according to a cost function, as the result for a given object. The cost function can be a sum of absolute differences between motion compensated pixels from neighbouring images, with vectors generated with the (Candidate) motion model. However, we need to know the area to which the motion model is to be assigned. The two issues, segmentation and motion estimation, are interdependent. In order to correctly estimate the motion in one object, the object should be known and vice versa.

As a first step in the motion estimation process we define a set with pixel blocks of interest. These form the set, $SI(n)$ of "interesting" image parts that will be used as a basis for optimization of *all* parametric models.

Now the focus of the individual parameter estimators has to be on different objects. To this end, each parameter estimator PE_o will calculate its cost function on the *same set* of interesting locations defined in set SI, giving different locations a *different weight factor*. $W_o(\vec{X})$. The proposed algorithm is straightforward:

- The pixel values are multiplied with a first weight factor larger than 1, e.g. 8, in case the pixel in $SI(n)$ belonged to the same object, i.e. the same parameter estimator, according to the previous image segmentation step.
- The pixel values are multiplied with a second weight factor smaller than 1, e.g. 0.1, in case the

following rule:

$$P(\vec{X},n) = \begin{cases} P_s & ,(M(\vec{X}+\vec{\delta},n) = o) \\ P_t & ,(M(\vec{X}-\vec{\delta},n-1) = o) \\ P_u & ,(otherwise) \end{cases} \quad (8)$$

and

$$\vec{\delta} = \begin{pmatrix} i \\ j \end{pmatrix}, i,j = 0,\pm 1 \quad (9)$$

Similar to what has been suggested for the 3-D RS block matcher, P_u is the largest penalty, P_t just a small one, while there is no reason why P_s could not just be zero. A fairly obvious simplification is to fix $\vec{\delta}$ to the direction opposite to the scanning direction, and to alternate the scanning from field to field. Figure 3 gives an example of a segmentation according to the *object-based* motion estimation method, with the original luminance image.

References

[1] G. de Haan, R.J. Schutten, and A. Pelagotti. Motion estimation and motion compensated interpolation, Philips Patent Application, PHN16.529. WO 99/16251

[2] G. de Haan, 'Judder-free video on PCs'. *Proc. of the WinHEC'98*, Mar. 1998, Orlando, (CD-ROM).

[3] G. de Haan and P.W.A.C. Biezen, 'An efficient true-motion estimator using candidate vectors from a parametric motion model'. *IEEE tr. on Circ. and Syst. for Video Techn.*, Vol. 8, no. 1, Mar. 1998, pp. 85-91.

[4] R.J. Schutten and G. de Haan, 'Real-time 2-3 pull-down elimination applying motion estimation / compensation on a programmable device', *IEEE Tr. on Consumer Electronics*, Vol. 44, No. 3, Aug. 1998, pp. 930-938.

[5] G. de Haan, P.W.A.C Biezen, H. Huijgen, and O.A. Ojo, 'True Motion Estimation with 3-D Recursive Search Block-Matching', *IEEE Tr. on Circuits and Systems for Video Technology*, Vol. 3, October 1993, pp. 368-388.

[6] G. de Haan, and P.W.A.C. Biezen, 'Sub-pixel motion estimation with 3-D recursive search block-matching', *Signal Processing: Image Communication 6*, 1994, pp.229-239. G. de Haan, J. Kettenis, and B. Deloore, 'IC for Motion Compensated 100Hz TV, with a Smooth Motion Movie-Mode'. *IEEE Tr. on Consumer Electronics*, vol. 42, no. 2, May 1996, pp. 165-174.

[7] G. de Haan and P.W.A.C. Biezen, 'Time-recursive de-interlacing for high-quality television receivers,' *Proc. of the Int. Workshop on HDTV and the Evolution of Television*, Taipei, Taiwan, November 1995, pp. 8B25–8B33.

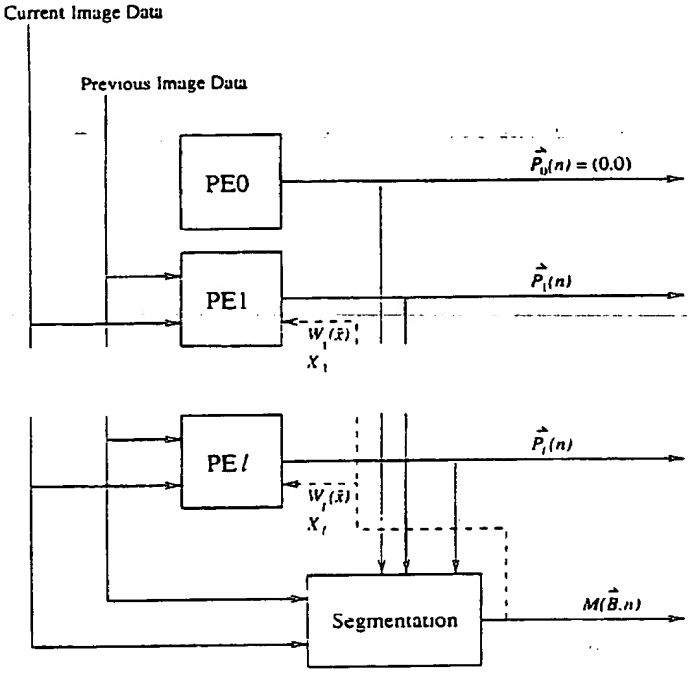

Figure 1: Block diagram of multiple parameter estimators and segmentation.

$$\vec{D}_o(\vec{x},n) = \begin{pmatrix} s_x(o,n) + x d_x(o,n) \\ s_y(o,n) + y d_y(o,n) \end{pmatrix} \qquad (1)$$

using $\vec{D}_o(\vec{x},n)$ for the displacement vector of object o at location $\vec{x} = \begin{pmatrix} x \\ y \end{pmatrix}$ in the image with index n.

1.2 Parameter estimation

Given a motion model, next its parameters need to be optimised for a given object in the image. As stationary image parts occur in almost every sequence, we assume the presence of an 'object o, $o = 0$', for which motion is described by, $\vec{0}$, the zero vector. Clearly no estimation effort is required to make this available. The parameter vectors of additional objects, o, $o > 0$, are estimated separately, in parallel, by their respective parameter estimators (PE_o), as shown in Figure 1. Each PE_o updates a previously estimated parameter vector, after which the best parameter candidate vector, according to a cost function, is selected as the result parameter vector for that object.

Considering the four parameter model of eq. (1), the parameters of object o, $o > 0$, are regarded as a parameter vector $\vec{P}_o$:

$$\vec{P}_o(n) = \begin{pmatrix} s_x(o,n) \\ s_y(o,n) \\ d_x(o,n) \\ d_y(o,n) \end{pmatrix} \qquad (2)$$

and we define our task as to select $\vec{P}_o(n)$ from a number of candidate parameter vectors $\vec{C}_o(n)$ as the one that has the minimal value of a cost function, to which we shall return lateron.

The candidates are generated much similar to the strategy exploited in [?, ?], i.e. take a prediction vector, add at least one update vector, and select the best candidate parameter vector (eq. ??). Candidate parameter set $CS_o(n)$ contains three candidates $\vec{C}_o(n)$ according to:

$$CS_o(n) = \{\vec{C}_o(n) | \vec{C}_o(n) = \vec{P}_o(n-1) + m\vec{U}_o(n), \vec{U}_o(n) \in US_o(n), m = -1, 0, 1\} \qquad (3)$$

with update parameter $\vec{U}_o(n)$ selected from update parameter set $US_o(n)$:

$$US_o(n) = \left\{ \begin{pmatrix} i \\ 0 \\ 0 \\ 0 \end{pmatrix}, \begin{pmatrix} 0 \\ i \\ 0 \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ i \\ 0 \end{pmatrix} \begin{pmatrix} 0 \\ 0 \\ 0 \\ i \end{pmatrix} \right\}, \quad (i = 1, 2, 4, 8, 16) \qquad (4)$$

1.3 The cost function

Given the motion model and some candidate parameter sets, we need to select the best candidate, according to a cost function, as the result for a given object. The cost function can be a sum of absolute differences between motion compensated pixels from neighbouring images, with vectors generated with the (Candidate) motion model. However, we need to know the area to which the motion model is to be assigned. The two issues, segmentation and motion estimation, are interdependent. In order to correctly estimate the motion in one object, the object should be known and vice versa.

As a first step in the motion estimation process we define a set with pixel blocks of interest, or feature points. These form the set, $SF(n)$ of "features" that will be used as a basis for optimization of *all* parametric

the image. since inconsistencies, similar to those of a full search block matcher, are not prevented.

Therefore, we adopted a segmentation method that more closely resembles the strategy of the 3-D RS block matcher, i.e. use spatial and temporal predictions of the best $PE_o(n)$ and penalise chosing a $PE_o(n)$ that does not occur in the spatio-temporal neighbourhood. Formally. the segmentation mask $M(\vec{X}, n)$ assigns the object o with the lowest local *modified* cost function $\epsilon_o{}'(\vec{X}, n)$ to the block $B(\vec{X})$, where

$$\epsilon_o{}' = \epsilon_o + P(\vec{X}, n) \tag{7}$$

while $P(\vec{X}, n)$ is a penalty choosen according to the following rule:

$$P(\vec{X}, n) = \begin{cases} P_s & ,(M(\vec{X} + \vec{\delta}, n) = o) \\ P_t & ,(M(\vec{X} - \vec{\delta}, n-1) = o) \\ P_u & ,(otherwise) \end{cases} \tag{8}$$

and

$$\vec{\delta} = \begin{pmatrix} i \\ j \end{pmatrix}, i, j = 0, \pm 1 \tag{9}$$

Similar to what has been suggested for the 3-D RS block matcher, P_u is the largest penalty, P_t just a small one, while there is no reason why P_s could not just be zero. A fairly obvious simplification is to fix $\vec{\delta}$ to the direction opposite to the scanning direction, and to alternate the scanning from field to field. Figure 3 gives an example of a segmentation according to the *object-based* motion estimation method, with the original luminance image.

2 Selection of the feature points

As a first step in the motion estimation process we defined a set with "features" (see section 1.3). This set, $SF(n)$ of "interesting" image parts is used as a basis for optimization of *all* parametric models.

An advantegeous way to obtain this set, *to be used in the parameter estimation of the next picture pair*. is to fill it with all blocks (on the sub-sampled image grid) that don't match well (threshold) with the corresponding blocks in the previous image:

$$SF(n) = \left\{ \vec{X} | \epsilon_0(\vec{X}, n-1) \geq Th \right\} \tag{10}$$

where $\epsilon_0 = \epsilon_o$ for $o = 0$.

This set, because of the threshold, contains no flat areas, as these would not help in the minimization of the cost function and therefore unnecessarily increase the operation count. Furthermore, the algorithm eliminates all stationary image parts from feature set, even if they contain high contrast details. That is another reduction of the number of irrelevant image parts, since the parameter "estimator" generating the "zero velocity model" requires no optimization on pixel data. In other words: the set of features contains blocks with significant detail in non-stationary image parts. exactly the best basis for all parameter estimators for non-stationary objects.

The threshold can be determined with a control loop that keeps the size of the set more or less constant. As an alternative. a relatively small fixed threshold can be used, such that always more than enough image parts are contained in an initial set. In this case. the final set results by sub-sampling the initial set with a factor such that a more or less constant number of image parts results.

References

[1] G de Haan. R.J. Schutten. and A. Pelagotti. Motion estimation and motion compensated interpolation. Philips Patent Application. PHN16.529. WO 99/16251

[2] G de Haan. 'Judder-free video on PCs'. *Proc. of the WinHEC'98*. Mar. 1998. Orlando. (CD-ROM).

[3] G de Haan and P.W.A.C. Biezen. 'An efficient true-motion estimator using candidate vectors from a parametric motion model'. *IEEE tr. on Circ. and Syst. for Video Techn.*. Vol. 8. no. 1. Mar 1998. pp. 85-91.

[4] G. de Haan. P.W.A.C. Biezen. and R.J. Schutten. Motion estimation. Philips Patent Application. PHN16.112. WO 97/46022

[5] R.J. Schutten and G. de Haan. 'Real-time 2-3 pull-down elimination applying motion estimation / compensation on a programmable device'. *IEEE Tr on Consumer Electronics*. Vol. 44. No. 3. Aug. 1998. pp. 930-938.

[6] G. de Haan. P.W.A.C Biezen. H. Huijgen. and O.A. Ojo. 'True Motion Estimation with 3-D Recursive Search Block-Matching'. *IEEE Tr on Circuits and Systems for Video Technology*. Vol. 3. October 1993. pp. 368-388.

[7] G. de Haan. and P.W.A.C. Biezen. 'Sub-pixel motion estimation with 3-D recursive search block-matching'. *Signal Processing: Image Communication 6*. 1994. pp 229-239 G. de Haan. J. Kettenis. and B. Deloore. 'IC for Motion Compensated 100Hz TV. with a Smooth Motion Movie-Mode'. *IEEE Tr on Consumer Electronics*. vol. 42. no. 2. May 1996. pp. 165-174.

A "meta model" improving an object motion estimator

G. de Haan and R.J. Wittebrood

Philips Research Laboratories, Television Systems Group,
Prof. Holstlaan 4, 5656 AA Eindhoven, The Netherlands

A "meta model" is proposed for an earlier designed object based motion estimator, suitable for high-quality motion compensated scan-rate conversions. The meta model is a parametric model, with a larger number of parameters than the models used to describe the objects in the object motion estimator, with its parameters derived from the output vector field of this object motion estimator, and acting as a candidate object in the segmentation stage of the object motion estimator. Compared to the prior art, the advantage of the resulting object motion estimator with this meta model is that an accurate model of complex global motion can be added as a powerful candidate to improve the output vector field, with a relative small calculational effort.

Keywords: **motion estimation, parametric motion models, image segmentation, scan rate conversion.**

1 Introduction

The motion estimator described in this paper was designed to be suitable for scan rate conversion, with a computational complexity suitable for consumer electronics application, i.e. comparable to [6, 7].

The most striking characteristic of the object motion estimator described earlier in [1], is that no effort is put in segmenting the image into objects prior to estimation of the model parameters. Basically, a relatively small number of interesting image parts is selected, and a number of parallel motion model parameter estimators is trying to optimize their parameters on this data set. As soon as one of the estimators is more successful than another in a certain number of interesting image parts, it is focused on those parts, whereas the remaining estimators focus on the other parts. In short: individual estimators try to conquer image parts from eachother, dividing the total image into "objects".

Fundamentally, such an *object-based* motion estimator[1] that wastes no effort in expensive segmentation of the image should be able to compete in operations count with a *block based* motion estimator, as one should expect less objects than blocks in realistic images. It is only in the assignment of image parts to objects that an effort is required comparable to the evaluation of candidate vectors on block basis. If the number of objects does not exceed the number of candidate vectors too much, the overhead of an object based motion estimator should be negligible. It is assumed here that the motion per object can be described with fairly simple parametric models(, corrected with a local bias vector, according to an earlier invention disclosure).

In this disclosure, a "meta model" is proposed to improve this object motion estimator. The meta model is a parametric model, with a larger number of parameters than the models used to describe the objects in the object motion estimator, with its parameters derived from the output vector field of this object motion estimator, and acting as a candidate object in the segmentation stage of the object motion estimator. Compared to the prior art, the advantage of the resulting object motion estimator with this meta model is that an accurate model of complex global motion can be added as a powerful candidate to improve the output vector field, with a relative small calculational effort. The proposed improvement is related to the improvement of the 3-D RS block matcher with a candidate vector derived from a parametric motion model, as has been documented in [3, 4].

In the following subsections, we shall describe the motion model used in the prior art object motion estimator, the estimation of the model parameters, the cost function used, the recursive segmentation of the im-

[1] An object, in the context of this estimator, refers to image parts that can be described with the same motion model. These image parts do not necessarily correspond to a single physical object in the scene.

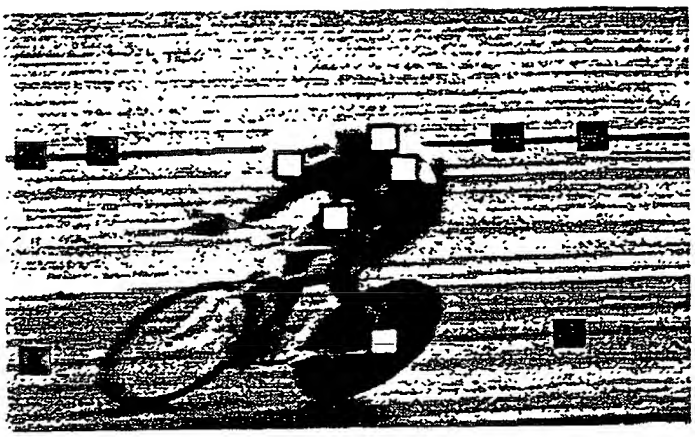

Figure 2: Example of selection of points of interest. The grey blocks are emphasized pixel blocks of interest in PE_1, the white ones are emphasized in PE_2.

to a cost function, as the result for a given object. The cost function can be a sum of absolute differences between motion compensated pixels from neighbouring images, with vectors generated with the (Candidate) motion model. However, we need to know the area to which the motion model is to be assigned. The two issues, segmentation and motion estimation, are interdependent. In order to correctly estimate the motion in one object, the object should be known and vice versa.

As a first step in the motion estimation process we define a set with pixel blocks of interest. These form the set, $SI(n)$ of "interesting" image parts that will be used as a basis for optimization of *all* parametric models.

Now the focus of the individual parameter estimators has to be on different objects. To this end, each parameter estimator PE_o will calculate its cost function on the *same set* of interesting locations defined in set SI, giving different locations a *different weight factor*, $W_o(\vec{X})$. The proposed algorithm is straightforward:

- The pixel values are multiplied with a first weight factor larger than 1, e.g. 8, in case the pixel in $SI(n)$ belonged to the same object, i.e. the same parameter estimator, according to the previous image segmentation step.
- The pixel values are multiplied with a second weight factor smaller than 1, e.g. 0.1, in case the segmentation assigned the position to another parameter estimator and this estimator achieved low match errors.

Figure 2 gives an example of the selection of the pixel blocks of interest in an image with a single moving object and a moving background. The moving background of the image is object $o = 1$, and the bicyclist is object $o = 2$. Both parameter estimators are optimized on the same set containing the blocks of interest, but as soon as one estimator is selected in the segmentation to be best in an area, the pixel block of interest in that area is emphasized in the cost function. After a while, this converges to the situation illustrated, where one estimator focusses om the grey blocks and the other on the white pixel blocks in $SI(n)$.

More formally, the cost function is calculated according to:

$$\epsilon(\vec{C}_o, n) = \sum_{\vec{x} \in IS} W_o(\vec{x}) \cdot |F_s(\vec{x}, n) - F_s(\vec{x} - \vec{C}_o(\vec{x}, n), n-1)| \quad (5)$$

where $F_s(\vec{x}, n)$ is the luminance value at position $\vec{x}$ in a *sub-sampled* image with index n, and $C_o(\vec{x}, n)$ is the vector resulting from candidate model $\vec{C}_o(n)$ at position $\vec{x}$.

The sub-sampling effectively reduces the required memory bandwidth. Images are sub-sampled with a factor of four horizontally and a factor of two vertically on a field base, generating a sub-sampled image $F_s(n)$ from each original field $F(n)$. In order to achieve pixel accuracy on the original pixel grid of F, interpolation is required on the sub-sampling grid.

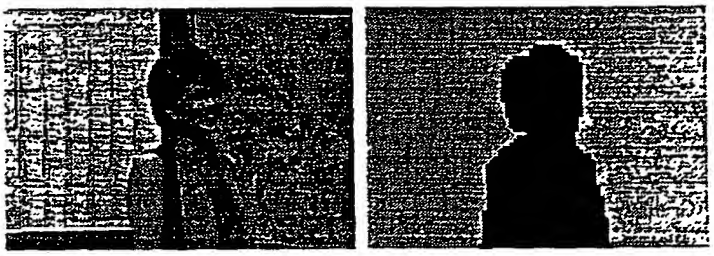

Figure 3: Picture and the resulting segmentation.

1.4 Recursive segmentation

The segmentation is the most critical step in the algorithm. Its task is to assign one motion model to each group of pixels. This is basically achieved by assigning the best matching model to each group of pixels (a block B, which is typically as large as 8×8 pixels on frame base).

For each block, a *block* match error, $\epsilon_o(\vec{X}, n)$ corresponding to each of the estimated parameter vectors.

compensation on a programmable device', *IEEE Tr. on Consumer Electronics*, Vol. 44. No. 3. Aug 1998, pp. 930-938.

[6] G. de Haan. P.W.A.C Biezen. H. Huijgen. and O.A. Ojo, 'True Motion Estimation with 3-D Recursive Search Block-Matching', *IEEE Tr. on Circuits and Systems for Video Technology*, Vol. 3. October 1993. pp. 368-388.

[7] G. de Haan, and P.W.A.C. Biezen, 'Sub-pixel motion estimation with 3-D recursive search block-matching', *Signal Processing: Image Communication 6*, 1994, pp.229-239. G. de Haan. J. Kettenis. and B. Deloore. 'IC for Motion Compensated 100Hz TV. with a Smooth Motion Movie-Mode'. *IEEE Tr. on Consumer Electronics*. vol. 42. no. 2. May 1996, pp. 165-174.

G. de Haan and P.W.A.C. Biezen. 'Time-recursive de-interlacing for high-quality television receivers.' *Proc. of the Int. Workshop on HDTV and the Evolution of Television*. Taipei. Taiwan. November 1995. pp. 8B25–8B33.

Claims

Claim 1: A method, and apparatus realizing this method, for estimating motion from video data. comprizing:

- generating at least two motion parameter models
- furnishing candidate parameter models for every picture part (block);
- assigning either of the candidate parameter models to blocks, using a criterion function:
- furnishing a bias field, describing local deviations from the selected motion parameter model

Claim 2: A method, and apparatus realizing this method. according to claim 1, in which the aforementioned segmentation is a recursive segmentation. i.e. selects the motion parameter model for the current block from a set containing motion models selected in a spatio-temporal neighbourhood, plus at least one additional motion model.

Claim 3: A method. and apparatus realizing this method, according to the previous claims, in which the bias field is generated according to the following steps:

- fetch at least one prediction bias value from a spatio-temporal neighbourhood;
- add a bias update value to at least one of the bias prediction values;
- assign the best bias value, according to a criterion function, to the current block

Claim 4: A method, and apparatus realizing this method, according to any of the previous claims, in which the criterion function mentioned in claim 3 is a match error criterion, e.g. a sum of absolute pixel differences, calculated between a block in the current picture and a block in the previous picture shifted over a motion vector resulting from the motion model selected for the current block and the candidate bias field value.

Claim 5 A method. and apparatus realizing this method, for segmenting pictures into areas (objects) the motion of which is described described by a single motion parameter model, comprising the steps of:

- furnishing (, for a group of pixels) of at least one (segmentation) prediction, taken from a spatio- temporal neighbourhood of the segmentation mask;
- selecting (, for that group of pixels) at least one additional segmentation candidate;
- deciding upon the segmentation (, for that group of pixels) using a criterion function.

Claim 6 A method, and apparatus realizing this method, according to claim 5, in which the aforementioned criterion function is a match error. e.g. a sum of absolute pixel differences.

Claim 7 A method, and apparatus realizing this method, according to the previous claims 5, 6, in which the match error for additional segmentation candidate is increased with a "penalty" value.

Claim 8 A method. and apparatus realizing this method. according to any of the previous claims. in which the processing order of the picture parts (blocks, or groups of pixels) is different for successive iterations, or successive fields.